

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	Surajit Chaudhuri	Attorney Docket No.:	160325.01
Application No.:	09/759,804	Group Art Unit	2163
Filed:	January 12, 2001	Examiner:	Filipczyk, M
Customer No.:	22971	Confirmation Number:	2731
Title: SAMPLING FOR QUERIES			

APPEAL BRIEF

To: Commissioner for Patents  
PO Box 1450  
Alexandria, Virginia 22313-1450

From: Microsoft Corporation  
Customer No. 22971

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits an appeal brief for application 09/759,804, filed January 12, 2001, with a request for a two month extension of time under 37 C.F.R. §41.37(e). Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<u>Appeal Brief Items</u>	<u>Page</u>
Real Party in Interest	3
Related Appeals and Interferences	3
Status of Claims	3
Status of Amendments	3
Summary of Claimed Subject Matter	4
Grounds of Rejection to be Reviewed on Appeal	5
Argument	5
Appendix of Appealed Claims	14
Evidence Appendix	17
Related Proceedings Appendix	18

**Real Party in Interest**

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject invention.

**Related Appeals and Interferences**

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

**Status of Claims**

Claims 49–54 stand rejected and are pending in the Application. Claims 49–54 are appealed. Claims 49–54 were previously amended. Claims 1–48 were previously canceled without prejudice. Claims 49–54 are set forth in the Appendix of Appealed Claims on page 23.

**Status of Amendments**

A Final Office Action was issued on December 7, 2006.

Appellant filed a Notice of Appeal on February 7, 2007 in response to the Final Office Action.

No claims have been amended since the issuance of the Final Office Action.

**Summary of Claimed Subject Matter**

The pending independent claims are claims 49 and 53. A concise explanation of each of the independent claims is provided below.

**Claim 49** describes a computer-implemented method comprising determining a weight for each tuple in a set of tuples based on a number of times the tuple was accessed during one or more past queries executed against the database (page 19, line 29 to page 20, line 3 and FIG. 5, 520, 530), selecting a subset of tuples from the set of tuples based on the determined weights (page 20, lines 4–5 and FIG. 5, 540), and determining an answer to a given query using the subset of tuples (page 20, lines 6–7 and FIG. 5, 550).

**Claim 53** describes a computer-readable media storing computer-executable instructions when executed by a computing device implement a method comprising collecting information related to one or more queries against a database, examining the collected information to determine a number of times each tuple in a set of tuples in the database was accessed during the one or more past queries (page 19 line 25–28 and FIG. 5, 510), determining a weight for each tuple in the set of tuples based on the number of times the tuple was accessed (page 19, line 29 to page 20, line 3 and FIG. 5, 520, 530), selecting a subset of tuples from the set of tuples based on the determined weights (page 20, lines 4–5 and FIG. 5, 540), and executing a query against the subset of tuples to determine an answer to the query made against the database (page 20, lines 6–7 and FIG. 5, 550).

**Grounds of Rejection to be Reviewed on Appeal**

Claims 49–51 and 53 stand rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,026,391 to Osborn et al (hereinafter “Osborn”).

**Argument**

- I. The rejection under 35 U.S.C. §102(e) over Osborn is improper because Osborn does not anticipate each and every element of the claims.

Claims 49–51 and 53 stand rejected under 35 U.S.C. §102(e) as being anticipated by Osborn.

Applicant respectfully submits that the Office has not established a proper anticipation rejection because Osborn does not anticipate each and every element of the claims.

- A. **The §102 Standard**

In making out a §102 rejection, the Federal Circuit has stated that “[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegall Bros. v. Union Oil Co. of California*, 814, F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Furthermore, “[t]he identical invention must be shown in as complete detail as is contained in the...claim....” *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

**B. Summary of Disclosure in Osborn**

Osborn discloses a method and apparatus for providing an estimate of the elapsed time required for a computer system to respond to queries. In particular, Osborn discloses a query performance prediction ("QPP") module that correlates estimated system cost information provided for each new query with statistics compiled from previous queries in order to estimate the time required by the system to respond to the query.

The estimated system cost information is generated by a cost optimizer based on the data distribution and storage characteristics for respective tables, clusters and indexes to be used for a set of potential execution plans for a given SQL query statement. The statistics compiled from previous queries are collected and maintained in a query history. The query history includes a query ID, the date and time at which the query was executed, an identifier of the user that executed the query, an identifier of the result set including the table(s) and column(s) accessed as part of the query, a Boolean indicating if the query was satisfied from a pre-computed summary table, the estimated cost for the query, the actual CPU time used in running the query, and the estimated time and the actual total elapsed time required for the system to complete the query.

The query performance prediction module estimates the elapsed time required for a given query by comparing the cost estimate, as generated by the cost optimizer described above, and result set for the given query to the recorded estimated costs and result sets of past queries stored in the query history cache. In the event an exact match is found, the query performance module selects the recorded CPU time for the matching past query as an

estimated CPU time for the given query. If an exact match is not found, a nearest neighbor algorithm is used to extrapolate an estimated CPU time based on a weighted average of CPU times for the closest matched queries stored in the query history cache.

**C. Summary of Key Differences Between Application and Osborn**

The disclosure of Osborn is primarily concerned with estimating the elapsed time a query requires to execute. In contrast, the present application is concerned with estimating or determining the answer to a query. That is, the product of Osborn is a single number representing the time estimate drawn either from an exact match or nearest neighbor match of the query under investigation. In contrast, the answer to a query includes any number of tuples selected from database tables in accordance with the structure of the query itself.

**D. Claims 49 and 53**

**Claim 49** is directed to computer-implemented method comprising determining a weight for each tuple in a set of tuples, the determination based on a number of times the tuple was accessed during one or more past queries executed against the database. Claim 49 is further directed to selecting a subset of tuples from the set of tuples based on the determined weights, and using the selected subset of tuples to determine an answer to a given query.

Claim 53 is directed to a computer-readable media storing computer executable instructions which when executed by a computing device implement a method comprising collecting information related to one or more past queries against a database and examining the collected information to determine a number of times each tuple in a set of tuples has been accessed during the one or more past queries. Claim 53 is further directed to determining a weight for each tuple in a set of tuples, the determination based on a number of times the tuple was accessed. Finally, Claim 53 is directed to selecting a subset of tuples from the set of tuples based on the determined weights, and executing a query against the selected subset of tuples to determine an answer to the query.

The Office refers to several locations in Osborn in asserting that Osborn teaches all of the claim limitations of Claim 49 and 53. The Office asserts that “examining the collected information to determine an access frequency for each tuple accessed during the one or more past queries” is taught at col. 6, lines 51–54, 59, and 65–67 of Osborn, noting “when a query is submitted only once the tuple in the database may be accessed once” (see page 3, Final Office Action issued on December 7, 2006).

The cited section of Osborn does not teach “examining the collected information to determine an access frequency for each tuple accessed during the one or more past queries”. Instead, the cited section of Osborn teaches that “each user station records pertinent information for each new user query, which are collected and maintained by the host computer in a query history”. Osborn only teaches that information is collected and does not teach that the recorded

information is examined, let alone that the information is examined to determine an access frequency for each tuple.

Furthermore, the cited section of Osborn does not discuss tuples in the database. The Office draws an equivalency between a tuple and tables and columns in the database in asserting that an access frequency for each tuple is equivalent to “particular table and columns accessed” (see Final Office Action of December 7, 2007, page 3). A tuple is not equivalent to a particular table and column; rather, a tuple maps a field name to a specific value. In particular, a tuple exists as a unique field within a column, and a column exists as a unique collection of fields within a database table.

More particularly, when a specific tuple is selected as part of a query, it is not inherent that the entire column or table that includes the specific tuple is also accessed. Therefore, even if Osborn were to disclose a determination of an access frequency of particular tables and column, which Osborn does not, it is not inherent that an access frequency for each tuple will also be determined.

Regardless, the Office asserts Osborn inherently discloses determining an access frequency because the access frequency of a specific tuple will always be a value of “1”, noting, “when a query is submitted only once the tuple in the database may only be accessed once” (see Final Office Action of December 7, 2007, page 3). However, consider that a tuple may be accessed any number of times during execution of a query. For example, consider a compound query with the first part of the query intended to select a first value from a column and a second part of the query intended to select a second value from the column. In order to perform the first part of the query and select the first value, the first

part of the query accesses the first value to select it and accesses the second value to determine it is not to be selected. Accordingly, the second part of the query then accesses the second value for a second time to select the second value and then accesses the first value for a second time to determine it is not to be selected.

The Office then refers to the statistics cache of Osborn in asserting that “determining sample weight based on the access frequency of the tuples” is disclosed at col. 7, lines 17–23 of Osborn (see Final Office Action of December 7, 2006, page 3). However, the cited section of Osborn fails to disclose determining sample weight based on the access frequency of the tuples and instead discloses that the query performance prediction (QPP) module, having not found an exact match between a given query and a query stored in the query statistics cache, then executes a “nearest neighbor” algorithm to extrapolate an estimated CPU time based on a weighted average of CPU times for the closest matching stored queries.

Regardless, Osborn does not disclose determining the access frequency of the tuples as previously discussed. Therefore, even if the cited section of Osborn were to disclose a “sample weight”, which Osborn does not, Osborn can not disclose determining a sample weight based on the access frequency of the tuples. In particular, Osborn does not disclose an access frequency upon which to base a sample weight.

The Office then cites items 45, 46, and 48 of FIG. 2 in asserting that Osborn discloses selecting a sample of tuples from the database based on the weights of the tuples (see Final Office Action of December 7, 2006, page 3).

Item 45 of FIG. 2 of Osborn is titled "QUERY RESULT SET", Item 46 of FIG. 2 of Osborn is titled "QPP MODULE", and Item 48 of FIG. 2 of Osborn is titled "QUERY STAT. CACHE". The Office notes that "the time, date and exact records, i.e., particular table and column access is stored in cache 48, lines 6, lines 51–68". It is assumed that the Office intended to cite col. 6, lines 51–68 of Osborn. The cited section of Osborn only discloses the contents of the query history 50, but does not disclose any tuples, weights of tuples, let alone selecting a sample of tuples based on the weights of the tuples.

Finally the Office refers to FIG. 2 and 4, item 46, 54, and 80 in asserting that Osborn teaches "executing a query against the sample to determine an approximate answer to the query made against the database" (see Final Office Action of December 7, 2006, page 3). The Office further cites col. 7, lines 7–10 of Osborn, which read, "...the QPP module 46 compares 80 the cost estimate 44 and result set 45 for the present query to the recorded estimated costs and result sets of past queries stored in the query statistics cache 48" (emphasis added). The cited section of Osborn teaches that the QPP module compares a CPU time cost estimate to recorded CPU time estimated costs. The comparison of Osborn is not equivalent to the execution of a query, regardless if Osborn were to teach a sample of selected tuples based on weights, which Osborn also fails to teach. And such a failure is to be expected as Osborn is concerned with estimating the CPU time a given query will require to execute, not with determining an approximate answer to the given query.

For at least the above-identified reasons, applicant respectfully submits that Claims 49 and 53 are not anticipated by Osborn and are each allowable. Accordingly, the rejection of Claim 49 and of Claim 53 should be withdrawn.

E. Claims 50-51

Claims 50-51 depend from claim 49 and are allowable at least by virtue of that dependency. Accordingly, the rejection of Claims 50 and 51 should be withdrawn.

F. Claims 52 and 54

Claims 52 and 54 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Osborn in view of U.S. Patent No. 6,519,604 to Acharya et al. (hereafter "Acharya"). Claim 52 depends from Claim 49 and Claim 53 depends on Claim 52 and are allowable at least by virtue of that dependency. Accordingly, the rejection to Claims 52 and 54 should be withdrawn.

**Conclusion**

The Office's basis and supporting rationale for the § 102(e), 102(b) and 103(a) rejections is not supported by the disclosure of the cited references. Applicant respectfully requests that the rejections be overturned and that the pending claims be allowed to issue.

Respectfully Submitted,

Dated: May 7, 2007

By: /Stephen C. Siu/

Stephen C. Siu  
Microsoft Corp.  
Reg. No. 48,303  
One Microsoft Way,  
Redmond, WA 98052  
Peter Taylor

**CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]**

I hereby certify that this correspondence and the documents identified on this form are being electronically deposited with the USPTO via EFS-Web on the date shown below:

May 7, 2007  
Date

/Kate Marochkina/  
Kate Marochkina

Appendix of Appealed Claims

49. A computer-implemented method, comprising:

determining a weight for each tuple in a set of tuples in a database based on a number of times the tuple was accessed during one or more past queries executed against the database;

selecting a subset of tuples from the set of tuples based on the determined weights;

determining an answer to a given query using the subset of tuples.

50. The method of claim 49, wherein determining the answer to the given query comprises:

executing the given query against the subset of tuples to identify answer tuples;

and

determining the answer to the given query using information from the answer tuples and the weights of the answer tuples.

51. The method of claim 49, wherein determining the answer to the given query comprises:

determining a probability with which each tuple in the subset of tuples was selected;

executing the given query against the subset of tuples to identify answer tuples;

and

determining the answer to the given query using information from the answer tuples and the probabilities of the answer tuples.

52. The method of claim 49, wherein determining the answer to the given query comprises:

determining a probability with which each tuple in the subset of tuples was selected;

executing the given query against the subset of tuples to identify answer tuples; and

determining the answer to the given query using information from the answer tuples and inverses of the probabilities of the answer tuples.

53. A computer-readable media storing computer-executable instructions, which, when executed by a computing device, implement a method comprising:

collecting information related to one or more past queries against a database;

examining the collected information to determine a number of times each tuple in a set of tuples in the database was accessed during the one or more past queries;

determining a weight for each tuple in the set of tuples based on the number of times the tuple was accessed;

selecting a subset of tuples from the set of tuples based on the determined weights;

executing a query against the subset of tuples to determine an answer to the query made against the database.

54. The computer-readable media of claim 53, wherein executing the query against the subset of tuples to determine the answer to the query made against the database comprises:

for each tuple in the subset of tuples, determining a probability with which the tuple was selected;

executing the query against the subset of tuples to identify answer tuples; and

determining the answer using information from the answer tuples in the subset of tuples and inverses of the probabilities of the answer tuples.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None